

NAME

csvconvert, csvsplit – facilities for parsing Comma-Separated-Values (CSV) data with gawk.

SYNOPSIS

```
@include "csv"
CSVFS = ...
CSVCOMMA = ...
CSVQUOTE = ...
result = csvconvert(csvrecord, option...)
n = csvsplit(csvrecord, afield, option...)
result = csvunquote(csvfield, option)      (see NOTE 1)
```

DESCRIPTION

The *csv* gawk extension adds functions for parsing CSV data in a simple way. The predefined CSVFS, CSVCOMMA and CSVQUOTE variables set default values for the optional arguments.

CSVFS

The field delimiter used in the resulting clean text record, initialized to a null character '\0'.

CSVCOMMA

The default field delimiter of the CSV input text, initialized to comma ','.

CSVQUOTE

The default quoting character of the CSV input text, initialized to double quote '"

csvconvert(*csvrecord* [, *fs* [, *comma* [, *quote*]])

Returns the CSV formatted string argument converted to a regular awk record with fixed field separators. Returns a null string if *csvrecord* is not a valid string. The arguments are as follows:

csvrecord

The CSV formatted input string

fs

The resulting field separator. Default CSVFS.

comma

The input CSV field delimiter. Default CSVCOMMA.

quote

The input CSV quoting character. Default CSVQUOTE.

csvsplit(*csvrecord*, *afield* [, *comma* [, *quote*]])

Splits the CSV formatted string argument into an array of individual clean text fields and returns the number of fields. Returns -1 if *csvrecord* is not a valid string. The arguments are as follows:

csvrecord

The CSV formatted input string

afield

The resulting array of fields.

comma

The input CSV field delimiter. Default CSVCOMMA.

quote

The input CSV quoting character. Default CSVQUOTE.

csvunquote(*csvfield* [, *quote*])

Returns the clean text value of the CSV string argument. Returns a null string if *csvfield* is not a valid string. The arguments are as follows:

csvfield

The CSV formatted input string

quote

The input CSV quoting character. Default CSVQUOTE.

EXAMPLES

Process CSV input records as arrays of fields:

```
{
    csvsplit($0, fields)
```

```
    if (fields[2]=="some value") print
}
Process CSV input records as awk regular records:
BEGIN {FS = "\\0"}
{
    CSVRECORD = $0
    $0 = csvconvert($0)
    if ($2=="some value") print CSVRECORD
}
```

BUGS

Null characters are not allowed in fields. A null character terminates the record processing.

AUTHOR

Manuel Collado, m-collado@users.sourceforge.net.

SEE ALSO

CSV Processing With gawk, csvmode(3am).

COPYING PERMISSIONS

Copyright (C) 2018, Free Software Foundation, Inc.

Permission is granted to make and distribute verbatim copies of this manual page provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual page under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual page into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.